```
PPPPPPPPPPP    LLL                        IIIIIIIII    RRRRRRRRRRR    TTTTTTTTTTTTTTT    LLL
PPPPPPPPPPP    LLL                        IIIIIIIII    RRRRRRRRRRR    TTTTTTTTTTTTTTT    LLL
PPPPPPPPPPP    LLL                        IIIIIIIII    RRRRRRRRRRR    TTTTTTTTTTTTTTT    LLL
PPP       PPP  LLL                           III       RRR       RRR        TTT          LLL
PPP       PPP  LLL                           III       RRR       RRR        TTT          LLL
PPP       PPP  LLL                           III       RRR       RRR        TTT          LLL
PPP       PPP  LLL                           III       RRR       RRR        TTT          LLL
PPP       PPP  LLL                           III       RRR       RRR        TTT          LLL
PPPPPPPPPPP    LLL                           III       RRRRRRRRRRR          TTT          LLL
PPPPPPPPPPP    LLL                           III       RRRRRRRRRRR          TTT          LLL
PPPPPPPPPPP    LLL                           III       RRRRRRRRRRR          TTT          LLL
PPP            LLL                           III       RRR   RRR            TTT          LLL
PPP            LLL                           III       RRR   RRR            TTT          LLL
PPP            LLL                           III       RRR   RRR            TTT          LLL
PPP            LLL                           III       RRR       RRR        TTT          LLL
PPP            LLL                           III       RRR       RRR        TTT          LLL
PPP            LLLLLLLLLLLLLLL            IIIIIIIII     RRR       RRR        TTT          LLLLLLLLLLLLLLL
PPP            LLLLLLLLLLLLLLL            IIIIIIIII     RRR       RRR        TTT          LLLLLLLLLLLLLLL
PPP            LLLLLLLLLLLLLLL            IIIIIIIII     RRR       RRR        TTT          LLLLLLLLLLLLLLL
```

```
PPPPPPPP   LL           IIIIII    CCCCCCCC  LL              000000    SSSSSSSS  EEEEEEEEEE
PPPPPPPP   LL           IIIIII    CCCCCCCC  LL              000000    SSSSSSSS  EEEEEEEEEE
PP     PP  LL             II    CC          LL            00     00   SS        EE
PP     PP  LL             II    CC          LL            00     00   SS        EE
PP     PP  LL             II    CC          LL            00     00   SS        EE
PPPPPPPP   LL             II    CC          LL            00     00   SSSSSS    EEEEEEE
PPPPPPPP   LL             II    CC          LL            00     00   SSSSSS    EEEEEEE
PP         LL             II    CC          LL            00     00        SS   EE
PP         LL             II    CC          LL            00     00        SS   EE
PP         LL             II    CC          LL            00     00        SS   EE          ....
PP         LLLLLLLLL    IIIIII    CCCCCCCC  LLLLLLLLL       000000    SSSSSSSS  EEEEEEEEEE   ....
PP         LLLLLLLLL    IIIIII    CCCCCCCC  LLLLLLLLL       000000    SSSSSSSS  EEEEEEEEEE   ....
```

```
LL           IIIIII    SSSSSSSS
LL           IIIIII    SSSSSSSS
LL             II    SS
LL             II    SS
LL             II    SS
LL             II      SSSSSS
LL             II      SSSSSS
LL             II           SS
LL             II           SS
LL             II           SS
LL             II           SS
LLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLL    IIIIII    SSSSSSSS
```

PLI$CLOSE
1-004
        - pl1 runtime close file
H 14
16-SEP-1984 02:11:51   VAX/VMS Macro V04-00   Page 1
6-SEP-1984 11:36:29 [PLIRTL.SRC]PLICLOSE.MAR;1   (1)

```
0000     1              .title  pli$close - pl1 runtime close file
0000     2              .ident  /1-004/                            ; Edit DSB1004
0000     3                                                         ; Edit CGN1003
0000     4                                                         ; Edit WHM1002
0000     5
0000     6
0000     7   ;*********************************************************************
0000     8   ;*                                                                   *
0000     9   ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                          *
0000    10   ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.           *
0000    11   ;*  ALL RIGHTS RESERVED.                                             *
0000    12   ;*                                                                   *
0000    13   ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000    14   ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
0000    15   ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
0000    16   ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000    17   ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP CF  THE  SOFTWARE IS  HEREBY *
0000    18   ;*  TRANSFERRED.                                                      *
0000    19   ;*                                                                   *
0000    20   ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
0000    21   ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
0000    22   ;*  CORPORATION.                                                      *
0000    23   ;*                                                                   *
0000    24   ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
0000    25   ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.          *
0000    26   ;*                                                                   *
0000    27   ;*                                                                   *
0000    28   ;*********************************************************************
0000    29   ;
0000    30
0000    31
0000    32   ;++
0000    33   ; facility:
0000    34   ;
0000    35   ;     VAX/VMS PL1 runtime library.
0000    36   ; abstract:
0000    37   ;
0000    38   ;     This module contains the pl1 runtime routine for closing a file.
0000    39   ;
0000    40   ; author: c. spitz 18-jul-79
0000    41   ;
0000    42   ; modified:
0000    43   ;
0000    44   ;
0000    45   ;     1-002   Bill Matthews   29-September-1982
0000    46   ;
0000    47   ;             Invoke macros $defdat and rtshare instead of $defopr and share.
0000    48   ;
0000    49   ;     1-003   Chip Nylander   23-February-1983
0000    50   ;
0000    51   ;             Add new optional parameter to PLI$CLOSE.  If parameter is
0000    52   ;             present, it specifies the severity level at which to signal
0000    53   ;             errors.
0000    54   ;
0000    55   ;     1-004   Dave Blickstein 26-March-1984
0000    56   ;
0000    57   ;             Change calls to LIB$FREE_VM to specify only the actual number
```

```
0000    58 ;                        of arguments they push (2) to the CALLS instruction.  The
0000    59 ;                        argument count was being used to pop local stack temporaries.
0000    60 ;                        This broke when new versions of the VM routines that took
0000    61 ;--                      optional arguments were called.
0000    62 ;--
0000    63
0000    64
0000    65 ;
0000    66 ; external definitions
0000    67 ;
0000    68
0000    69        $deffcb                              ;define file control block offsets
0000    70        $defpl1rtcons                        ;define pl1 runtime constants
0000    71        $defkcb                              ;define key control block offsets
0000    72        $defenv                              ;define environment block offsets
0000    73        $fabdef                              ;define fab offsets
0000    74        $namdef                              ;define nam offsets
0000    75
0000    76 ; local data
0000    77 ;
0000    78 ;
0000    79
0000    80        rtshare                              ;sharable
0000    81
0000    82
```

```
                                        0000        84
                                        0000        85   ;++
                                        0000        86   ; pli$close -- close a file
                                        0000        87   ;
                                        0000        88   ; functional description:
                                        0000        89   ;
                                        0000        90   ; This routine closes a pl1 file.  The syntax of a close statement is:
                                        0000        91   ;     CLOSE FILE(file-ref);
                                        0000        92   ;
                                        0000        93   ; inputs:
                                        0000        94   ;     (ap) = number of arguments (1)
                                        0000        95   ;     4(ap) = address of the fcb for this file
                                        0000        96   ;     8(ap) = address of the environment option
                                        0000        97   ;     12(ap) = severity level at which to signal errors
                                        0000        98   ;
                                        0000        99   ; outputs:
                                        0000       100   ;     the fcb is updated to reflect the closed status of the file.
                                        0000       101   ;     an rms close is issued, the file block is deallocated, and the
                                        0000       102   ;     fcb is removed from the open fcb list.
                                        0000       103   ;--
                                        0000       104
                             00FC       0000       105           .entry   pli$close,^m<r2,r3,r4,r5,r6,r7>
                                        0002       106   ;
                                        0002       107   ; check argument
                                        0002       108   ;
                        6C   D5         0002       109           tstl     (ap)                          ;if no parameters
                        0C   14         0004       110           bgtr     10$                           ;then
                        51   D4         0006       111           clrl     r1                            ;indicate no fcb
     50   00000000'8F   D0         0008       112           movl     #pli$_parm,r0                 ;set invalid parameters
                        0097 31         000F       113           brw      50$                           ;and fail
                                        0012       114
                                        0012       115   ; if file opened, close it.
                                        0012       116
        52    04 AC    D0         0012       117   10$:    movl     4(ap),r2                      ;get address of fcb
     03 0C A2    01    E0         0016       118           bbs      #atr_v_opened,fcb_l_attr(r2),20$ ;if file opened
                     010C 31         001B       119           brw      110$                          ;then
           00020002 8F    CA         001E       120   20$:    bicl     #<atr_m_opened!atr_m_bfall>,- ;set file not opened and
                        0C A2         0024       121                    fcb_l_attr(r2)                ;buffer not allocated
        53    00A6 C2    9E         0026       122           movab    fcb_b_fab(r2),r3              ;get address of fab
     0C A2    00001040 8F    D3         002B       123           bitl     #<atr_m_record! -             ;if record
                                        0033       124                    atr_m_input>,fcb_l_attr(r2)  ;or input
                        1F   12         0033       125           bneq     30$                           ;then cont
        05 0C A2    07    E1         0035       126           bbc      #atr_v_print,fcb_l_attr(r2),25$ ;if not print, cont
                     34 A2    D5         003A       127           tstl     fcb_l_prn(r2)                 ;any carriage control pending?
                        07   12         003D       128           bneq     26$                           ;if neq, yes, print it
     1C A2    14 A2    D1         003F       129   25$:    cmpl     fcb_l_buf(r2),fcb_l_buf_pt(r2) ;if buf empty
                        0E   13         0044       130           beql     30$                           ;then cont
                        5C   DD         0046       131   26$:    pushl    ap                            ;save ap
           5C   52    D0         0048       132           movl     r2,ap                         ;set fcb addr in ap
     00000000'GF   16         004B       133           jsb      g^pli$$put_rec                ;put the last record
                     5C 8ED0         0051       134           popl     ap                            ;restore ap
        02    6C   D1         0054       135   30$:    cmpl     (ap),#2                       ;close environment?
                        36   19         0057       136           blss     37$                           ;if lss, no
        08 AC    D5         0059       137           tstl     8(ap)                         ;anything there?
                        31   13         005C       138           beql     37$                           ;if eql, no
        5E   22    C2         005E       139           subl     #env_k_len,sp                 ;get room for env block
        56   52    7D         0061       140           movq     r2,r6                         ;save fcb and fab addr
```

```
       6E   22   00   6E   00   2C   0064   141              movc5    #0,(sp),#0,#env_k_len,(sp) ;zero env block
                     52   56   7D   006A   142              movq     r6,r2                     ;restore fcb and fab
       6E   00000040 8F   C8   006D   143              bisl     #env_m_close,env_l_status(sp) ;set close status
                          5E   DD   0074   144              pushl    sp                        ;set addr of env block
                     08 AC   DD   0076   145              pushl    8(ap)                     ;set addr of close environment
                          52   DD   0079   146              pushl    r2                        ;set addr of fcb
                     08 AE   9F   007B   147              pushab   8(sp)                     ;set addr of addr
                     08 AE   9F   007E   148              pushab   8(sp)                     ;set addr of addr
                     08 AE   9F   0081   149              pushab   8(sp)                     ;set addr of addr
       00000000'GF   06   FB   0084   150              calls    #6,g^pli$$envir           ;process environment
                     01 50   E8   008B   151              blbs     r0,37$                    ;if lbs, cont
                          04   008E   152              ret                                ;error already signaled, just return
                          008F   153 37$:         $close   r3                        ;close the file
                     34 50   E8   0098   154              blbs     r0,60$                    ;if error on close
       50   00000000'8F   D0   009B   155              movl     #pli$_rmsf,r0             ;then set rms fab error code
                     08 A2   50   D0   00A2   156 40$:         movl     r0,fcb_l_error(r2)        ;put error in fcb
                     51   52   D0   00A6   157              movl     r2,r1                     ;set fcb address
                          51   DD   00A9   158 50$:         pushl    r1                        ;push fcb addr
                          50   DD   00AB   159              pushl    r0                        ;push error code
       00000000'8F   DD   00AD   160              pushl    #pli$_error               ;set error condition
                     03   6C   D1   00B3   161              cmpl     (ap),#3                   ;error level specified?
                     06   19   00B6   162              blss     53$                       ;if lss, no
       6E   03   00   0C AC   F0   00B8   163              insv     12(ap),#0,#3,(sp)         ; set severity to specified level
       00000000'GF   03   FB   00BE   164 53$:         calls    #3,g^pli$io_error         ;signal the condition
                          04   00C5   165              ret                                ;return
       50   00000000'8F   D0   00C6   166 55$:         movl     #pli$_virmemdeal,r0       ;set vir. mem. deallocation error
                     D3   11   00CD   167              brb      40$                       ;and fail
                          00CF   168 ;
                          00CF   169 ; deallocate file buffer
                          00CF   170 ;
                     14 A2   D5   00CF   171 60$:         tstl     fcb_l_buf(r2)             ;if buffer allocated
                     27   13   00D2   172              beql     80$                       ;then
       06 0C A2   0C   E0   00D4   173              bbs      #atr_v_record,fcb_l_attr(r2),70$ ;if stream
       7E   2A A2   3C   00D9   174              movzwl   fcb_w_linesize(r2),-(sp)  ;set length of buffer
                     03   11   00DD   175              brb      75$                       ;cont
                     18 A2   DD   00DF   176 70$:         pushl    fcb_l_buf_end(r2)         ;push length of buffer
                     14 A2   DF   00E2   177 75$:         pushal   fcb_l_buf(r2)             ;push address of buffer
                     04 AE   DF   00E5   178              pushal   4(sp)                     ;push address of length
       00000000'GF   02   FB   00E8   179              calls    #2,g^lib$free_vm          ;free buffer
                     5E   04   C0   00EF   180              addl2    #4,sp                     ;
                     D1 50   E9   00F2   181              blbc     r0,55$                    ;if deallocation error then fail
                     14 A2   D4   00F5   182              clrl     fcb_l_buf(r2)             ;indicate buffer not allocated
                     18 A2   D4   00F8   183              clrl     fcb_l_buf_end(r2)
       0101   C2   94   00FB   184 80$:         clrb     <fcb_b_nam+nam$b_esl>(r2) ;clear esl
       23 0C A2   10   E1   00FF   185              bbc      #atr_v_indexed,fcb_l_attr(r2),85$ ;if indexed
                     38 A2   D5   0104   186              tstl     fcb_l_kcb(r2)             ;kcb allocated?
                     1E   13   0107   187              beql     85$                       ;if eql, no, cont
       51   3C A2   9A   0109   188              movzbl   fcb_b_numkcbs(r2),r1      ;get number of kcbs
       7E   51   2C   C5   010D   189              mull3    #kcb_c_len,r1,-(sp)       ;get size of kcbs
                     38 A2   DF   0111   190              pushal   fcb_l_kcb(r2)             ;push address of kcbs
                     04 AE   DF   0114   191              pushal   4(sp)                     ;push address of length
       00000000'GF   02   FB   0117   192              calls    #2,g^lib$free_vm          ;free kcbs
                     5E   04   C0   011E   193              addl2    #4,sp                     ;
                     A2 50   E9   0121   194              blbc     r0,55$                    ;if error, fail
                     38 A2   D4   0124   195              clrl     fcb_l_kcb(r2)             ;indicate no kcbs
                          0127   196 ;
                          0127   197 ; remove file from open file queue, set it not open and return
```

```
                    0127   198 .
      52  62  0F    0127   199 85$:    remque  (r2),r2          ;remove fcb from queue
      50  01  D0    012A   200 110$:   movl    #1,r0            ;set success
              04    012D   201         ret                      ;return
                    012E   202
                    012E   203         .end
```

```
$$.TMP1                    = 00000001     KCB_W_LEN2              00000012
$$.TMP2                    = 00000053     KCB_W_LEN3              00000016
ATR_M_BFALL                = 00020000     KCB_W_LEN4              0000001A
ATR_M_INPUT                = 00000040     KCB_W_LEN5              0000001E
ATR_M_OPENED               = 00000002     KCB_W_LEN6              00000022
ATR_M_RECORD               = 00001000     KCB_W_LEN7              00000026
ATR_V_INDEXED              = 00000010     KCB_W_POS0              00000008
ATR_V_OPENED               = 00000001     KCB_W_POS1              0000000C
ATR_V_PRINT                = 00000007     KCB_W_POS2              00000010
ATR_V_RECORD               = 0000000C     KCB_W_POS3              00000014
ENV_K_LEN                    00000022     KCB_W_POS4              00000018
ENV_L_FILEIDTO               00000014     KCB_W_POS5              0000001C
ENV_L_FXCTLTO                00000018     KCB_W_POS6              00000020
ENV_L_STATUS                 0000000C     KCB_W_POS7              00000024
ENV_M_CLOSE                = 00000040     LIB$FREE_VM            ********   X   02
ENV_Q_CREDATE                00000004     NAM$B_ESC             = 0000000B
ENV_Q_EXPDATE                0000000C     PLIS$ENVIR            ********   X   02
ENV_W_OWNGROUP               0000001E     PLIS$PUT_REC          ********   X   02
ENV_W_OWNMEM                 00000020     PLISCLOSE             00000000  RG   02
ENV_W_PROT                   0000001C     PLI$IO_ERROR          ********   X   02
FCB_B_ENVIR                  000001C2     PLI$_ERROR            ********   X   02
FCB_B_ESA                    0000012E     PLIS_PARM             ********   X   02
FCB_B_EXTRA                  0000003D     PLIS_RMSF             ********   X   02
FCB_B_FAB                    000000A6     PLIS_VIRMEMDEAL       ********   X   02
FCB_B_IDENT                  00000040     SIZ...               = 00000001
FCB_B_IDENT_NAM              00000042     SYS$CLOSE             ********  GX   02
FCB_B_NAM                    000000F6
FCB_B_NUMKCBS                0000003C
FCB_B_RAB                    00000062
FCB_C_LEN                    000001C2
FCB_C_STRLEN                 00000034
FCB_L_ATTR                   0000000C
FCB_L_BUF                    00000014
FCB_L_BUF_END                00000018
FCB_L_BUF_PT                 0000001C
FCB_L_CNDADDR                000001B2
FCB_L_CONDIT                 000001AE
FCB_L_DTTR                   00000010
FCB_L_ERROR                  00000008
FCB_L_KCB                    00000038
FCB_L_NEXT                   00000000
FCB_L_PREVIOUS               00000004
FCB_L_PRN                    00000034
FCB_Q_RFA                    00000020
FCB_W_COLUMN                 0000002E
FCB_W_IDENT_LEN              00000040
FCB_W_LINE                   00000030
FCB_W_LINESIZE               0000002A
FCB_W_PAGE                   00000032
FCB_W_PAGESIZE               0000002C
FCB_W_REVISION               00000028
KCB_C_LEN                    0000002C
KCB_L_DTYP                   00000000
KCB_L_PREC                   00000004
KCB_L_ZERO                   00000028
KCB_W_LEN0                   0000000A
KCB_W_LEN1                   0000000E
```

```
                                    +-------------------+
                                    ! Psect synopsis !
                                    +-------------------+

PSECT name                     Allocation        PSECT No.  Attributes
----------                     ----------        ---------  ----------
. ABS .                        00000000 (    0.)  00 (   0.)  NOPIC  USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                          000001C2 (  450.)  01 (   1.)  NOPIC  USR   CON   ABS   LCL NOSHR  EXE   RD     WRT NOVEC BYTE
_PLI$CODE                      0000012E (  302.)  02 (   2.)    PIC  USR   CON   REL   LCL  SHR   EXE   RD   NOWRT NOVEC LONG

                              +-----------------------------+
                              ! Performance indicators !
                              +-----------------------------+

Phase                   Page faults   CPU Time       Elapsed Time
-----                   -----------   --------       ------------
Initialization               9        00:00:00.03    00:00:01.09
Command processing          72        00:00:00.47    00:00:06.16
Pass 1                     176        00:00:06.10    00:00:23.63
Symbol table sort            0        00:00:00.63    00:00:02.03
Pass 2                      41        00:00:01.05    00:00:04.54
Symbol table output          9        00:00:00.08    00:00:00.08
Psect synopsis output        2        00:00:00.02    00:00:00.02
Cross-reference output       0        00:00:00.00    00:00:00.00
Assembler run totals       309        00:00:08.38    00:00:37.55
```

The working set limit was 900 pages.
30745 bytes (61 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 516 non-local and 16 local symbols.
203 source lines were read in Pass 1, producing 13 object records in Pass 2.
18 pages of virtual memory were used to define 16 macros.

```
                              +-------------------------------+
                              ! Macro library statistics !
                              +-------------------------------+

Macro library name                          Macros defined
------------------                          --------------
_$255$DUA28:[PLIRTL.OBJ]PLIRTMAC.MLB;1             5
_$255$DUA28:[SYSLIB]STARLET.MLB;2                  8
TOTALS (all libraries)                            13
```

593 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=TRACEBACK/LIS=LIS$:PLICLOSE/OBJ=OBJ$:PLICLOSE MSRC$:PLICLOSE/UPDATE=(ENH$:PLICLOSE)+LIB$:PLIRTMAC/L

PLICASERR
LIS

PLICONDIT
LIS

DATDEF
SDL

FILEDEF
IN

PLIRTDEF
MAR

PLICHRSTR
LIS

PLICHAR
LIS

PLICLOSE
LIS

PLIRTL

PLIFILDSP
IN

PLIBYTSIZ
LIS

PLIRTL
MAP

ENVCODES
IN

PLIBIT
LIS